

Univerzální publikační systém pro malé a střední webové projekty

Universal Publishing System for Small and Middle Web Projects

Zadání bakalářské práce

Student: **Marek Záškodný**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Univerzální publikační systém pro malé a střední webové projekty**
Universal Publishing System for Small and Middle Web Projects

Zásady pro vypracování:

Potřeba prezentace na internetu je dnes stále důležitějším nástrojem v různých oblastech. Existuje celá řada různých publikačních systémů, které jsou více či méně dostupné, kvalitní a univerzální. Cílem tohoto projektu je vytvořit jednoduchý univerzální publikační systém, který bude využívat aktuální možnosti a trendy webových technologií.

1. Analyzujte a popište potřeby prezentace na internetu pro malé a střední webové projekty.
2. Navrhněte koncepci a architekturu publikačního systému, který bude zohledňovat specifika definovaná v předchozím bodě. Důraz řešení bude kladen na jednoduchost, flexibilitu a vícepoužitelnost výsledného řešení.
3. Implementujte publikační systém s využitím aktuálních moderních přístupů, nástrojů a technologií.
4. Vytvořte ukázkovou komplexní prezentaci postavenou na vyvinutém systému. Tvorbu ukázkové prezentace detailně popište formou manuálu.
5. Zhodnoťte vytvořené řešení a jeho možnosti praktického využití.

Seznam doporučené odborné literatury:

- [1] Ed Lecky-Thomson, Steven D. Nowicki: **PHP 6 - programujeme profesionálně**, 2010, Computer Press, ISBN: 978-80-251-3127-5
- [2] Bob Boiko: **Content Management Bible**, 2004, Wiley, ISBN: 978-0764573712

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Radecký**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli. Zejména svému vedoucímu bakalářské práce, panu Ing. Michalu Radeckému, Ph.D.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2012


.....

Abstrakt

V současné moderní době malé a střední firmy nemají šanci bez kvalitní webové prezentace uspět na trhu. Snahou této bakalářské práce je vytvoření jednoduchého univerzálního publikačního systému, který by těmto subjektům pomohl prezentovat se profesionálně na internetu. Tento systém by měl být srozumitelný i pro běžného uživatele, který nemá žádné znalosti o programování. K vytvoření webové prezentace uživateli stačí mít pouze základní dovednosti práce s počítačem. Důraz je tedy kladen na jednoduchost, flexibilitu, vícepoužitelnost a zároveň profesionalitu. Systém je vyvíjen za pomoci webových technologií a Nette Frameworku.

Klíčová slova: publikační systém, malé a střední firmy, webová prezentace, Nette Framework

Abstract

In our modern times small and middle companies do not have a chance without a good website to succeed in the market. The aim of this thesis is to create a simple universal publishing system, that would help these entities to present themselves professionally on the Internet. This system should be understandable for the common user, who has no knowledge of programming. To create a web presentation enough to have only basic computer skills. Emphasis is placed on simplicity, flexibility, more usability and also professionalism. System is developed using web technologies and Nette Framework.

Keywords: publishing system, small and middle companies, web project, Nette Framework

Seznam použitých zkratek a symbolů

AJAX	– Asynchronous JavaScript and XML
BCG	– Boston Consulting Group
CSS	– Cascading Style Sheets
CSRF	– Cross-Site Request Forgery
CMS	– Content Management System
FTP	– File Transfer Protocol
HTML	– HyperText Markup Language
LAMP	– Linux, Apache, MySQL a PHP, Perl nebo Python
MSSQL	– Microsoft SQL Server
MVC	– Model-View-Controller
MySQL	– My Structured Query Language
ORM	– Object-relational mapping
PHP	– Hypertext Preprocessor
SEO	– Search Engine Optimization
SŘBD	– Systém řízení báze dat
UML	– Unified Modeling Language
URL	– Uniform Resource Locators
UTF-8	– UCS Transformation Format
XHTML	– eXtensible HyperText Markup Language
XML	– eXtensible Markup Language
XSS	– Cross Site Scripting
WYSIWYG	– What you see is what you get
WWW	– World Wide Web

Obsah

1	Úvod	4
1.1	Prezentace firmy na internetu	4
2	Publikační systémy	7
2.1	Stručná historie	7
2.2	Příklady CMS	8
2.3	Publikační systém	10
3	Analýza systému	12
3.1	Diagram případů užití	12
3.2	Diagram aktivit	12
3.3	Minispecifikace	15
4	Návrh publikačního systému	18
4.1	Architektura	18
4.2	Návrh databáze	19
4.3	Bezpečnost	20
5	Popis implementace	22
5.1	Vlastní ovládací prvky pro formuláře	22
5.2	Stromová nabídka	25
5.3	Návštěvní kniha	26
5.4	Použití triggerů a transakcí pro mazání	27
5.5	Vzhled systému	30
6	Popis jednotlivých částí publikačního systému	32
6.1	Registrace a přihlášení	32
6.2	Správa projektu	32
6.3	Správa stránek	32
6.4	Hlavní části webové stránky	33
6.5	Správa Rozhraní	35
7	Případová studie ukázkové prezentace na reálném projektu	37
7.1	Porovnání vzhledu	37
7.2	Vlastní rozšířená prezentace	40
7.3	Zhodnocení ukázkových prezentací	42
8	Závěr	43
9	Literatura	44
10	Přílohy	45

Seznam obrázků

1	Statistika využití webových stránek	5
2	WordPress - ukázka administrace	8
3	Joomla - ukázka administrace	9
4	Diagram případů užití - správa systému	13
5	Diagram aktivit - správa projektu	14
6	Schéma databáze	20
7	Hlavní strana - registrace	30
8	Náhled administrace	31
9	Ukázka původní prezentace	38
10	Ukázka vlastní prezentace	39
11	Ukázka rozšířené prezentace - návštěvní kniha	40
12	Ukázka rozšířené prezentace - úvodní strana	41

Seznam výpisů zdrojového kódu

1	Minispecifikace - vytvoření nového projektu	15
2	Minispecifikace - stažení projektu	15
3	Ochrana formulářů	20
4	Šifrování hesel	21
5	Funkce SetValue pro DatePicker	22
6	Vložení komponenty DatePicker do presenteru	23
7	Vykreslení komponenty DatePicker	23
8	Nastavení JSColor inputu	23
9	Funkce pro odkazy v JavaScriptu	24
10	Funkce pro převod odkazu videa v PHP	24
11	Funkce pro převod odkazu mapy v PHP	24
12	Rekurzivní vykreslení levé nabídky	25
13	Rekurzivní vykreslení návštěvní knihy	26
14	Funkce pro reakce v návštěvní knize	27
15	Příkaz pro smazání Prvku	27
16	Trigger Prvku před smazáním	27
17	Trigger základního Rozhraní před smazáním	28
18	Trigger základního Rozhraní po smazání	28
19	Trigger Objektu	28
20	Transakce pro smazání levé nabídky	28
21	Trigger pro Spoj po smazání	29
22	Trigger pro pomocné tabulky	29
23	Šifrování kontrolního kódu	32

1 Úvod

Bakalářská práce se zabývá problematikou tvorby publikačního systému známějšího pod názvem CMS - systém pro správu obsahu. Cílem této práce je vytvoření systému, který by byl jednoduchý, univerzální a přitom co nejlépe pokryl potřeby malých a středních firem. Systém by měl tedy obsahovat hlavní prvky stránek jako jsou obsah, hlavička, patička, nabídka a jejich správu. Dále prvky pro vytvoření obsahu stránek. Např.: texty, obrázky, videa, odkazy, nadpisy, mapy, kalendáře... Výsledný webový projekt by měl být také SEO optimalizovaný. Další důležitou funkcí jsou statistiky přístupů.

Celá práce je rozdělena na několik částí. Nejprve je popsána potřeba prezentace pro firmy a způsoby tvorby CMS. Další část je zaměřena na publikační systém od návrhu architektury, databáze až po popis jednotlivých částí systému a implementace. Na závěr je představen navržený ukázkový webový projekt vytvořený pomocí publikačního systému a hodnocení výsledků práce.

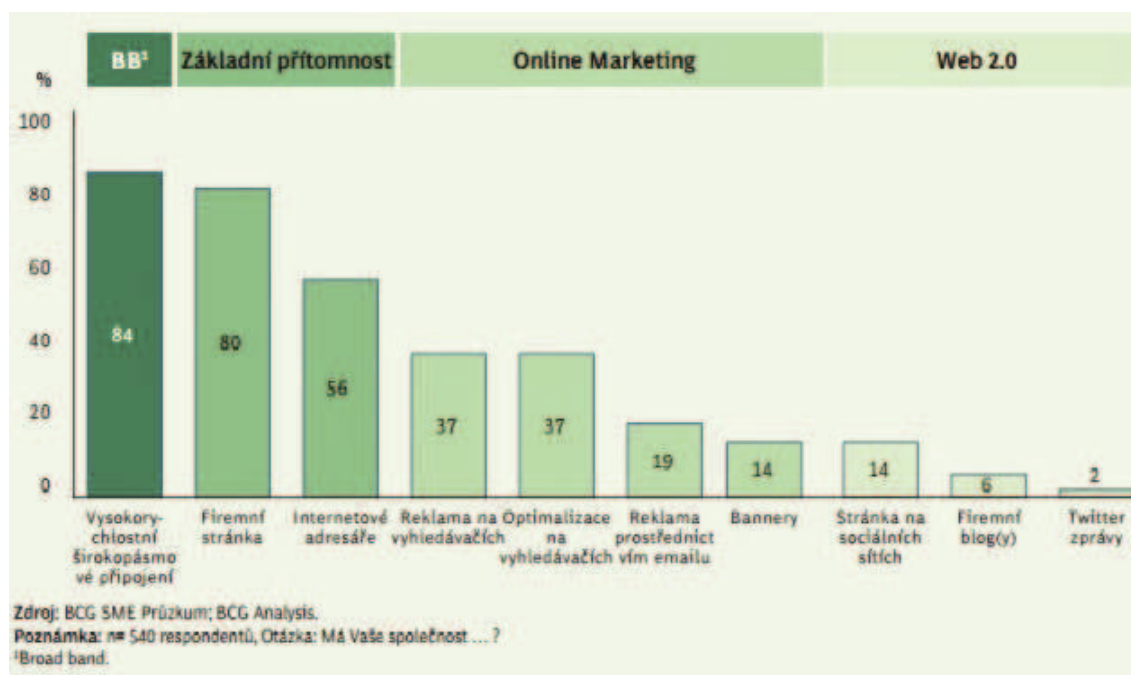
Ukázka publikačního systému se nachází na adrese: <http://zaskodny.cz/bakalarka/>.

1.1 Prezentace firmy na internetu

Rozvoj telefonních sítí byl základním prvkem pro vznik internetu. Internet vznikl počátkem 60.let minulého století ve Spojených státech amerických pro vojenské účely. Byl záměrně navržen tak, aby posílané informace neměly předem určenou cestu ani cíl. V podstatě stejně funguje i dnešní internet.

Na přelomu 80. a 90. let vzniká World Wide Web, zprvu pro podporu vědecké činnosti. Tato informační služba převzala navigační služby a spojuje v sobě hypertextové odkazy, dokumenty, text i grafické prvky. V současné době je celosvětová síť internetu v podstatě globálním činitelem, která díky internetovým překladačům překonává i jazykové bariéry. Na celém světě mají přístup k internetu asi 2 miliardy lidí a v ČR pak asi 6 až 7 milionů lidí. [1]

Situaci v prezentaci firemních webových stránek charakterizuje článek Hospodářských novin, (28.4.2011), vycházející ze studie společnosti Boston Consulting Group [2]. Studie uvádí, že 80% malých a středních firem již má své webové stránky, ale jen 37% využívá aktivně možnosti reklamy na internetu a optimalizace na vyhledávačích Google a Seznam. Graf se nachází na obrázku č. 1.



Obrázek 1: Statistika využití webových stránek

Dle indexu e-intenzity (zahrnujícího vlivy kvality internetové infrastruktury, dostupnosti připojení, výše nákupů spotřebitelů a firem přes internet a jak aktivně přijímají firmy, veřejný sektor a spotřebitelé internet) je ČR na 22.místě ze sledovaných 28 zemí z celého světa. To jednoznačně svědčí o velkém prostoru pro zlepšení využití internetu pro malé a střední podniky.

Jedním ze zdrojů studie provedené společností BCG byl průzkum provedený ve více než 500 malých a středních firmách, definovaných jako společnosti s méně než 250 zaměstnanci.

Z tohoto průzkumu, kromě již výše uvedených skutečností, vyplývá, že malým a středním podnikům využívajícím internet se daří lépe, vzrostl jim prodej, jsou odolnější v prostředí postiženém recesí. Ze závěrů průzkumu také vyplývá, že nejčastějším důvodem proč firmy nevyužívají internet více, je nedostatek pochopení obchodních výhod, které toto medium nabízí a časově příliš náročná aktualizace.

Rozvoj internetu poskytuje výhody malým a středním firmám rozšířením jejich možností v oblasti komunikace, pokrytí, výkonnosti a růstu obchodních aktivit. Získávají rovnocennější postavení i v konkurenci s většími firmami na trhu. To je významné v ČR, kde malé a střední firmy tvoří více než polovinu ročních příjmů země a zaměstnávají více než 60% zaměstnanců. Ne všechny firmy využívají své internetové prezentace stejně. Stále existuje skupina firem, které neinvestovaly do on-line marketingu a nenabízejí možnost nákupu svých produktů přes internet. Většina tvoří skupinu, která nabízí své služby a produkty prostřednictvím internetu a nabízí rovněž on-line objednávání.

Stále je zde však skupina firem, které z různých důvodů ještě internetové prezentace nevyužívají vůbec. Zvláště na tuto skupinu je zaměřeno vytvoření návrhu univerzálního publikačního systému.

2 Publikační systémy

V současné době se publikační systémy (CMS) staly velmi důležitou součástí webových prezentací. Vedení společností využívá výhod těchto systémů. Umožňují marketingovým oddělením publikovat informace právě tehdy, kdy je to třeba. CMS je definováno jako množina procesů a technologií, které podporují životní cyklus vývoje elektronických informací [3]. V dnešní době se jako CMS zpravidla chápou webové aplikace, někdy s případným doplňkovým programovým vybavením u klienta.

Mezi základní funkce CMS patří:

- tvorba, modifikace a publikace dokumentů zpravidla prostřednictvím webového rozhraní, často s využitím jednoduchého online WYSIWYG editoru nebo jednoduchého systému formátování textu (není nutná znalost HTML)
- řízení přístupu k dokumentům, zpravidla se správou uživatelů a přístupových práv
- správa diskusí či komentářů
- správa souborů či obrázků
- kalendářní funkce
- statistiky přístupů

2.1 Stručná historie

Počátky webové správy obsahu začínají v roce 1995 ve společnostech InterLeaf a CNET. Společnosti zaznamenaly potřebu publikace informací ve WWW a tak inovátorská společnost InterLeaf vytvořila systém Cyberleaf, který umožňoval uživatelům transformovat dokumenty vytvořené v textových editorech do jazyka HTML a také umožňoval vytváření odkazů mezi jednotlivými dokumenty. Zhruba ve stejnou dobu vytvořila společnost CNET web používající interní webový publikační systém, který byl patentován pod názvem PRISM. V roce 1997 společnost Vignette vytvořila produkt Tool Command Language, který byl nesrovnatelně lepší než tehdejší produkty. Společnosti si mohly tento produkt zakoupit pomocí licence. V průběhu dalších let se oblast správy obsahu zaplnila velice rychle novými společnostmi jako byly: BroadVision, InterWoven, Documentum, DynaBase, Future Tense, FileNet, Stellent a stovky dalších. Po propadu cen akcií na burze mezi roky 2000 až 2003 se trh správy obsahu rozdělil na dvě hlavní části [3]:

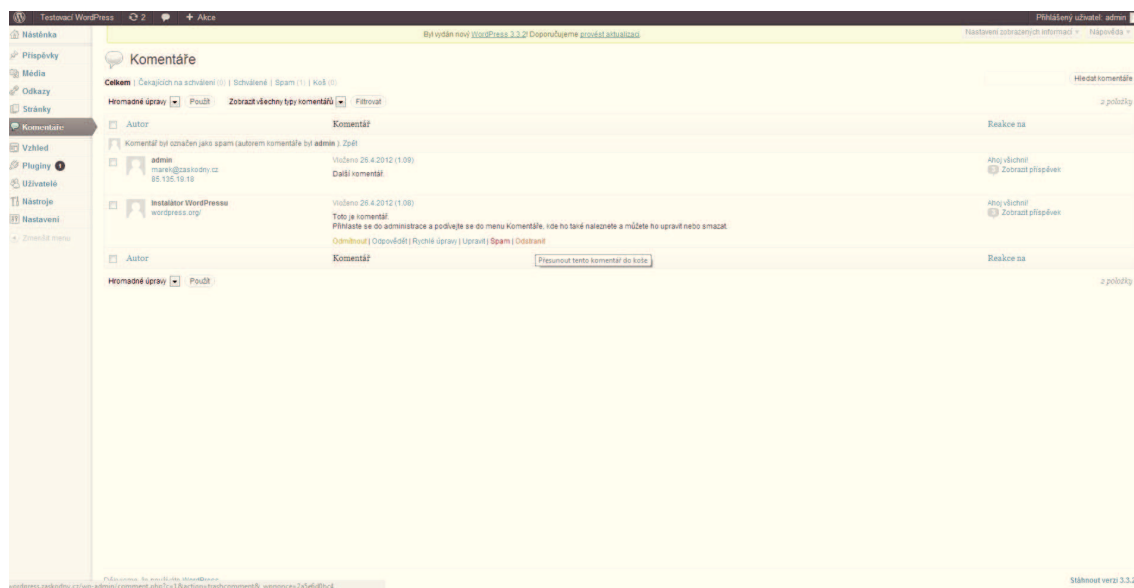
1. Velké společnosti jako IBM a Vignettes, které působily v enterprise prostředí. Jejich systémy zpravidla myslely ve velkém - databázový server Oracle. Pracovala na nich dlouhou dobu řada konzultantů a měly enormní náklady.
2. Open-source systémy, které jsou ve většině případů webového charakteru a používaly kombinaci označovanou zkratkou LAMP (Linux, Apache, MySQL, PHP). Některé byly i komerční (např.: Mambo). Mezi nejznámější patří: Joomla! (odnož Mambo), Drupal a WordPress.

Nové systémy CMS byly velmi dostupné nebo úplně zdarma. Mnoho z těchto menších systémů se navíc zaměřovalo na to, aby jejich cílová skupina uživatelů byla maximálně spokojená. Drupal původně vznikl jako softwarová vývěska. WordPress začínal jako platforma pro blogy. Oba tyto systémy se, ale postupem času změnil v plnohodnotný systém CMS.

2.2 Příklady CMS

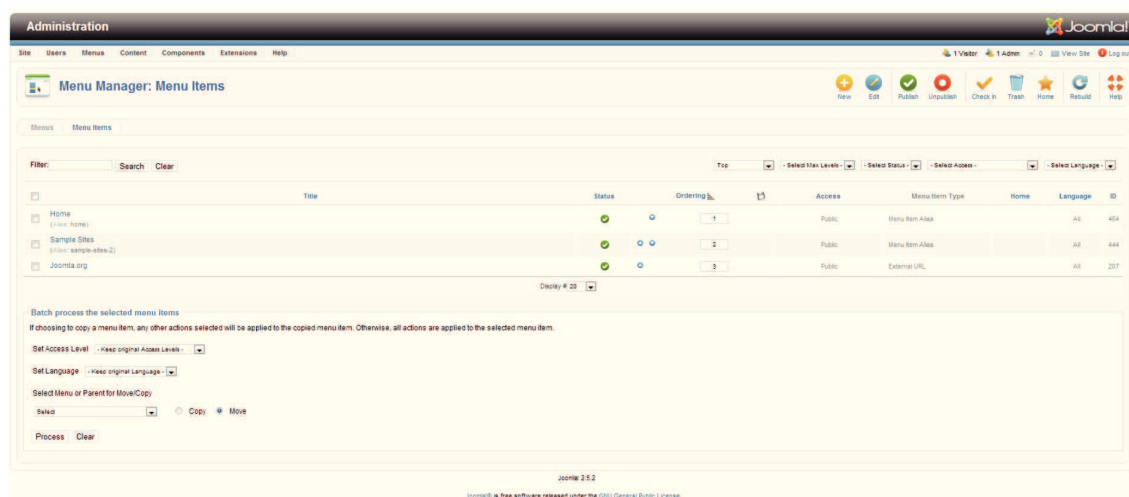
V předchozí kapitole je shrnuta historie CMS. Dále jsou uvedeny příklady nejznámějších publikačních systémů a jejich využití v dnešní době:

- **WordPress** - nejpoužívanější CMS a má širokou uživatelskou a vývojářskou komunitu. Používá jej více než 15% webových stránek na světě. Mnohé ze současných nejpoblárnějších blogů pohání WordPress. Například: TechCrunch, ICanHasCheezburger.com a New York Times. Dodržuje standardy XML, XHTML a CSS, podporuje pluginy a témata vzhledu, má integrovaného správce odkazů... [4] Ukázka administrace se nachází na obrázku č. 2.



Obrázek 2: WordPress - ukázka administrace

- **Drupal** - rozsáhlý systém pro správu obsahu. Veškerý obsah se ukládá do databáze, zpravidla MySQL. Kód propojující jednotlivé komponenty i moduly je vytvořen v PHP. Obsah se v systému ukládá ve formě tzv. uzlů. Tyto uzly mohou mít podobu článků nebo zpráv blogu a mohou mít vazby s jinými uzly. Ke všem uzlům jsou připojena univerzální metadata: identifikátor uzlu, typ obsahu, název, tělo, datum vytvoření a autor. Uzly je možné organizovat do kategorií. Drupal je schopný uspokojit požadavky pravidelné publikace článků díky standardním i doplňkovým modulům, může sloužit jako základ blogů, zpravodajských webů, firemních prezentací, linkovací web, katalog odkazů nebo jako osobní stránka. Nechybí ani řešení pro zdánlivě odlišné projekty jako jsou elektronické obchody nebo weby s nabídkami nemovitostí. Součástí Drupalu jsou také funkce pro vytvoření diskusního fóra. Na tomto systému běží například známý web <http://extrahardware.cz/>. [5]
- **Joomla!** - pracuje s články, které mohou představovat jakýkoliv typ obsahu. Články je možné zařazovat do kategorií a kategorie se řadí do sekcí. Dále umožňuje přidání dalších modulů, které zpravidla zajišťují specifické úkoly. Existují moduly pro zobrazení navigace, archivovaného obsahu, vlastní nabídky, zobrazení výsledků hlasování... Ukázka administrace se nachází na obrázku č. 3.



Obrázek 3: Joomla - ukázka administrace

2.3 Publikační systém

Při tvorbě publikačního systému jsou použity znalosti z výše zmíněných publikačních systémů a potřeby prezentace pro firmy, uvedené v první kapitole. Důraz řešení je kladen na jednoduchost a univerzálnost výsledného systému.

Celý systém se skládá z administrační části, kde se může návštěvník zaregistrovat a vytvářet si své vlastní projekty. Zde může své projekty spravovat a až budou hotové, tak je může umístit na vlastní webhosting. Druhou částí je samotná webová prezentace, kterou může z administrace vidět pomocí náhledu.

Pro jednoduchou správu projektu jsou stránky rozděleny na následující samostatné oddíly:

- **hlavička** - zde se může nahrát nadpis a obrázek
- **horní nabídka** - obsahuje pouze odkazy (lokální a nebo externí)
- **levá nabídka** - obsahuje odkazy
- **obsah** - jedná se o hlavní obsah webové stránky ve středu webové prezentace
- **pravá nabídka** - obsahuje odkazy
- **patička** - do patičky se může nahrát text

Povinnou součástí každé stránky projektu je pouze obsah. Ostatní části jsou pro jednotlivé stránky volitelné. Vzhled se volí pro každou část samostatně.

Hlavní obsah se v systému spravuje pomocí nadřazeného objektu. Existuje více typů objektu. Pro přehlednost se tento nadřazený objekt nazývá jako **Rozhraní**. Z důvodu univerzálnosti projektu je **Rozhraní** spravováno zvlášť níže pod projekty. Každé **Rozhraní** se dá vícenásobně použít pro neomezené množství projektů. Může se umístit do levé nabídky, obsahu a nebo pravé nabídky. **Rozhraní** funguje na podobném principu jako například uzly v Drupalu, které jsou zmíněny v předchozí kapitole.

Výslednou prezentaci je možno jednoduše stáhnout a umístit na vlastní webhosting. Samotný export projektu je popsán v tutoriálu (příloha č. 1) v levé nabídce. Jedná se o 4 jednoduché kroky:

1. **Systém** - je třeba stáhnout aktuální verzi systému v sekci **Ke Stáhnutí** a nahrát ji na FTP server webhostingu
2. **Databáze** - stáhnutí databázového souboru pro vytvoření tabulek v databázi a jeho následný import
3. **Projekt** - import záznamů do vytvořených tabulek v předchozím bodě
4. **Připojení k databázi** - nastavení souboru pro připojení k databázi

Podle návodu pro export by to měl během pár minut zvládnout i běžný uživatel.

Po nasazení projektu na nový webhosting je omezená administrace přístupná na adrese: <http://www.NAZEVDOMENY.cz/admin>. Přístupové údaje pro přihlášení jsou stejné jako v původní správě systému.

V celém projektu je kladen důraz na optimalizaci pro vyhledávače SEO [6] prostřednictvím vhodného použití titulků, nadpisů a zvýrazňování slov. Kód výsledné prezentace je HTML validní, což je také důležité. Pro SEO je nejdůležitější správné použití

titulků stránek. Z toho důvodu se titulek stránky volí pro každou stránku samostatně, aby co nejlépe odpovídal obsahu dané stránky.

V tutoriálu je také zmíněno, jak je pro SEO důležité mít správně nastavena klíčová slova a popis webového projektu.

3 Analýza systému

Další fází je analýza systému. Jedná se o převod slovního zadání do podoby vhodné pro vlastní implementaci projektu. Analýza obsahuje diagram případů užití (Use Case), diagram aktivit a minispifikaci.

3.1 Diagram případů užití

Tento diagram ukazuje jak funguje správa celého systému a jak probíhá vytváření nového projektu v rámci systému. Diagram případů užití se nachází na obrázku č. 4.

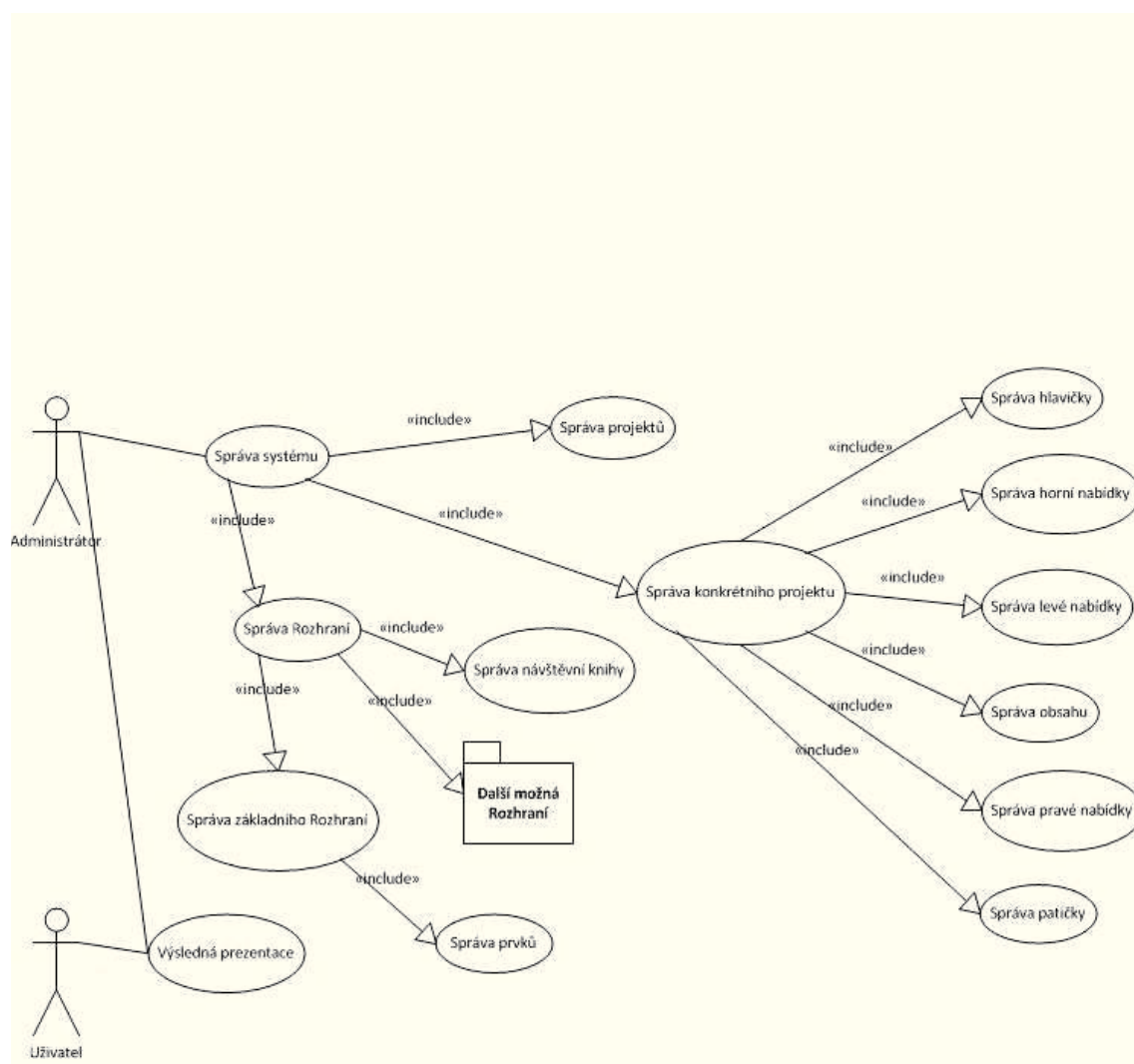
3.2 Diagram aktivit

Diagram aktivit je jedním z UML diagramů, který popisuje chování. Diagram aktivit je zobrazen na případu vytvoření projektu a následné práce s projektem. Nachází se na obrázku č. 5.

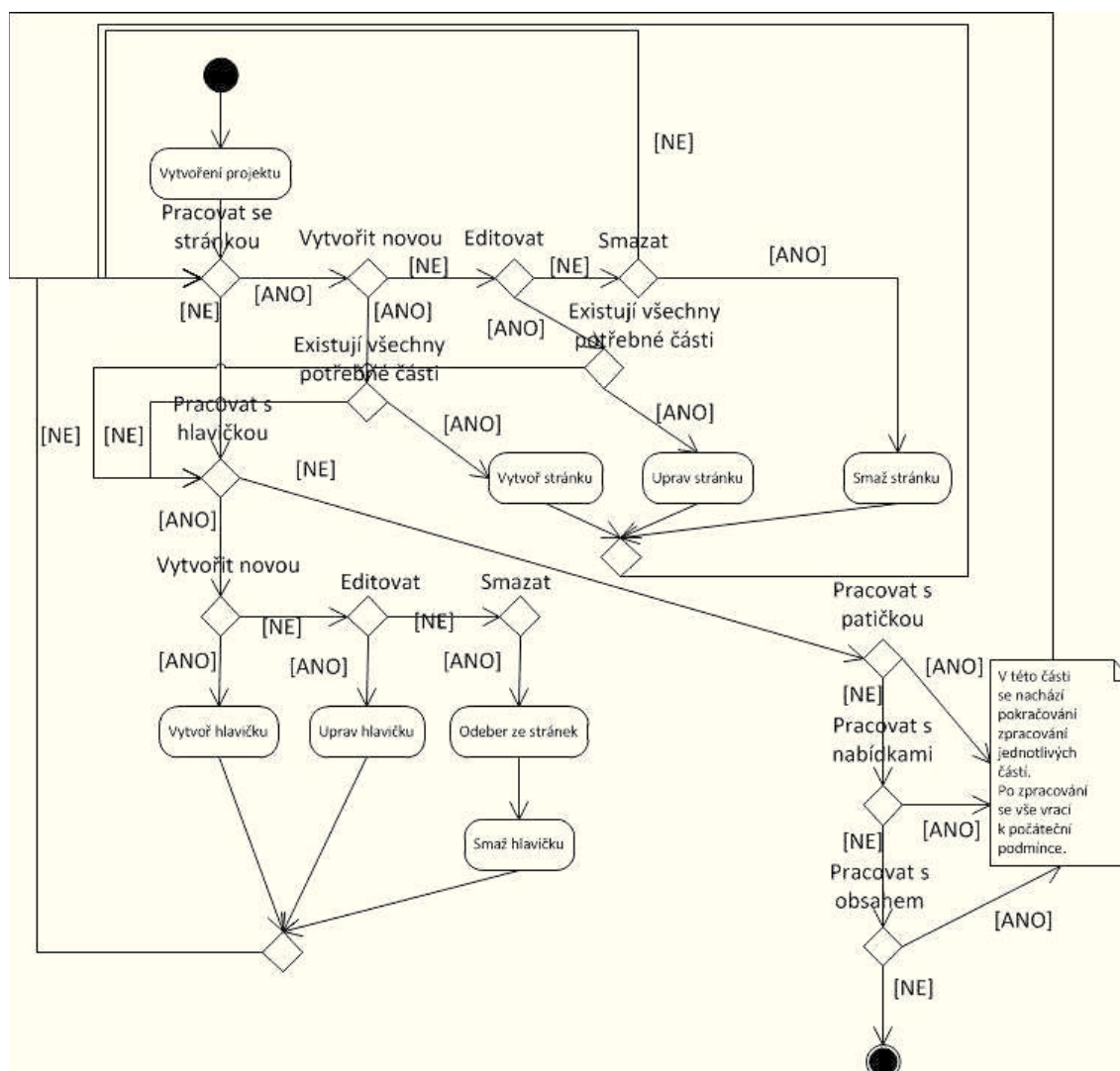
V diagramu je zobrazen způsob, jakým se tvoří nový webový projekt. Prvním krokem je samotné vytvoření projektu. Následuje tvorba vlastních stránek projektu. Při tvorbě stránek se vyberou části (hlavička, nabídky, obsah a patička), které daná stránka bude obsahovat. Pokud dané části neexistují tak se před samotným vytvořením stránek musí nejdříve vytvořit a poté přiřadit ke stránce.

Z jednotlivých částí stránky je pro ukázkou v diagramu zobrazena pouze práce s hlavičkou. Práce s dalšími částmi stránky je podobná, proto je pro lepší přehlednost tvorba dalších částí odkázána na poznámku v diagramu.

Z diagramu je dále vypuštěna práce s **Rozhraními**, které se vážou na levou a pravou nabídku a na obsah.



Obrázek 4: Diagram případů užití - správa systému



Obrázek 5: Diagram aktivit - správa projektu

3.3 Minispecifikace

Popisuje procesy na nejnižší úrovni. Následuje minispecifikace pro vytvoření nového projektu:

- 1) Uživatel vyplní název projektu
- 2) Vytvoř nové Rozhraní typu 1 (Vlastní), přiřaď k němu ID přihlášeného uživatele a pojmenuj ho zadaným názvem s přidaným textem "_objekt" na konci
- 3) Vytvoř v tabulce Hlavní nový záznam a přiřaď mu ID Rozhraní vytvořeného v předchozím bodě
- 4) Vytvoř nový prvek s 1 sloupcem, pořadí nastav na 1 a přiřaď mu ID předcházejícího záznamu
- 5) Vytvoř další prvek stejný jako předchozí až na pořadí, které nastav na 2
- 6) Vytvoř nový výchozí obrázek s popisem "Úvodní_text" a s ID prvního prvku
- 7) Vytvoř nový odstavec s výchozím textem a s ID druhého prvku
- 8) Vytvoř nový obsah s názvem ze zadání s přidaným textem "_obsah" na konci. Titulek nastav na "Můj_nový_projekt". Dále nastav výchozí barvy a ID Rozhraní na začátku
- 9) Vytvoř nový projekt pojmenovaný podle zadaného názvu a přiřaď mu ID přihlášeného uživatele.
- 10) Vytvoř novou stránku s názvem podle zadaného názvu s textem "_strana" na konci. Přiřaď stránce ID projektu a obsahu.
- 11) Uprav vytvořený obsah. Nastav mu ID vytvořeného projektu.
- 12) Uprav vytvořený projekt a nastav mu ID stránky.

Výpis 1: Minispecifikace - vytvoření nového projektu

Tímto postupem se vytvoří nový projekt s obsahem, který obsahuje 1 obrázek a 1 odstavec textu.

Dále následuje minispecifikace pro stažení celé webové prezentace, kterou si uživatel vytvořil v administraci:

- 1) Uživatel vybere projekt, který si chce stáhnout.
- 2) Najdi projekt podle zadaného ID od uživatele.
- 3) POKUD již existuje soubor s názvem vybraného projektu POTOM ho smaž
JINAK pokračuj
- 4) Najdi uživatele, kterému projekt patří
- 5) Vyber údaje uživatele z databáze a ulož je **do** proměnné
- 6) Najdi stránky, které patří k projektu
PRO KAŽDOU stránku PROVEĎ
- 7) Vyber údaje dané stránky a ulož je **do** proměnné
KONEC CYKLU
- 8) Najdi hlavičky, které patří k projektu
PRO KAŽDOU hlavičku PROVEĎ
- 9) Vyber údaje dané hlavičky z databáze a ulož je **do** proměnné
KONEC CYKLU
- 10) Najdi patičky, které patří k projektu
PRO KAŽDOU patičku PROVEĎ
- 11) Vyber údaje dané patičky z databáze a ulož je **do** proměnné
KONEC CYKLU
- 12) Vytvoř pole pro uložení záznamů z tabulky SPOJ, která slouží k navázání nabídek na položky menu
- 13) Najdi všechny horní nabídky, které patří k projektu
PRO KAŽDOU horní nabídku PROVEĎ
- 14) Vyber údaje dané horní nabídky z databáze a ulož je **do** proměnné
- 15) Přidej ID spoje horní nabídky **do** pole

KONEC CYKLU

16) Najdi všechny levé nabídky, které patří k projektu

PRO KAŽDOU levou nabídku PROVEĎ

17) Vyber údaje dané levé nabídky z databáze a ulož je **do** proměnné

18) Přidej ID spoje levé nabídky **do** pole

KONEC CYKLU

19) Najdi všechny pravé nabídky, které patří k projektu

PRO KAŽDOU pravou nabídku PROVEĎ

20) Vyber údaje dané pravé nabídky z databáze a ulož je **do** proměnné

21) Přidej ID spoje pravé nabídky **do** pole

KONEC CYKLU

PRO KAŽDÝ spoj z pole PROVEĎ

22) Najdi daný spoj

23) Vyber údaje daného spoje z databáze a ulož je **do** proměnné

24) Najdi všechny položky daného spoje

PRO KAŽDOU položku PROVEĎ

25) Vyber údaje dané položky z databáze a ulož je **do** proměnné

26) Zavolej rekurzivní funkci, která projde položky s potomky a uloží je **do** proměnné

KONEC CYKLU

KONEC CYKLU

27) Najdi všechny obsahy, které patří k projektu

PRO KAŽDÝ obsah PROVEĎ

28) Vyber údaje daného obsahu z databáze a ulož je **do** proměnné

KONEC CYKLU

29) Vytvoř pole pro uložení Rozhraní

30) Vyber všechna Rozhraní uživatele

PRO KAŽDÉ Rozhraní PROVEĎ

31) Vyber údaje daného Rozhraní z databáze a ulož je **do** proměnné

32) Přidej ID Rozhraní **do** pole

KONEC CYKLU

33) Vytvoř pole pro uložení základního Rozhraní

PRO KAŽDÉ ID z pole Rozhraní PROVEĎ

34) Najdi všechna základní Rozhraní podle ID Rozhraní

PRO KAŽDÉ základní Rozhraní PROVEĎ

35) Vyber údaje daného základního Rozhraní z databáze a ulož je **do** proměnné

36) Přidej ID základního Rozhraní **do** pole

KONEC CYKLU

37) Najdi všechny návštěvní knihy podle ID Rozhraní

PRO KAŽDOU návštěvní knihu PROVEĎ

38) Vyber údaje dané návštěvní knihy z databáze a ulož je **do** proměnné

KONEC CYKLU

KONEC CYKLU

39) Vytvoř pole pro uložení prvků základního Rozhraní

PRO KAŽDÉ ID z pole základního Rozhraní PROVEĎ

40) Najdi všechny prvky, které patří k ID základního Rozhraní

PRO KAŽDÝ prvek PROVEĎ

41) Vyber údaje daného prvku z databáze a ulož je **do** proměnné

42) Přidej ID prvku **do** pole

KONEC CYKLU

KONEC CYKLU

PRO KAŽDÉ ID z pole prvků PROVEĎ

43) Najdi všechny mezery, které patří k danému prvku

PRO KAŽDOU mezeru PROVEĎ

- 44) Vyber údaje mezery z databáze a ulož je **do** proměnné KONEC CYKLU
 - 45) Najdi všechny obrázky, které patří k danému prvku
PRO KAŽDÝ obrázek PROVEĎ
 - 46) Vyber údaje obrázku z databáze a ulož je **do** proměnné KONEC CYKLU
 - 47) Najdi všechny odstavce, které patří k danému prvku
PRO KAŽDÝ odstavec PROVEĎ
 - 48) Vyber údaje odstavce z databáze a ulož je **do** proměnné KONEC CYKLU
 - 49) Stejně pokračuj s dalšími prvky, které obsahuje systém KONEC CYKLU
 - 50) Zapiš údaje z proměnné pro ukládání výstupu databáze **do** souboru s názvem projektu
 - 51) Stáhni daný soubor
-

Výpis 2: Minispecifikace - stažení projektu

Vybrané údaje z databáze se ukládají do proměnné pro výstup ve formě INSERT dotazů pro MySQL databázi.

4 Návrh publikačního systému

V této části bakalářské práce se dostáváme již k samotnému návrhu systému. Uveden je zde popis architektury, databáze a také bezpečnost systému.

4.1 Architektura

Celý systém je programován v Nette Frameworku [8], který je založen na MVC přístupu. Knihovna hojně využívá návrhového vzoru convention-over-configuration. To znamená, že vývojář je co nejvíce odlehčen od zdoluhavého rozhodování a konfigurování, přičemž se však musí uspokojit s přednastavenou implementací.

4.1.1 Model

Model tvoří rozhraní pro práci s daty. Způsob implementace je volný a každý model je specifický pro jinou aplikaci.

4.1.2 View

Pohled (view) reprezentuje stav aplikace uživateli. Neměl by vyvolávat žádné změny v aplikaci. Není možné spravovat každou stránku samostatně. Z toho důvodu Nette nabízí řešení v podobě šablon. Jsou to dokumenty, které po zpracování zobrazí výslednou HTML stránku. Šablony zpracovává třída Template. Při vytváření instancí této třídy se jako parametr předává cesta k šabloně.

4.1.3 Controller

Řadič (controller) je jádro aplikace, které obsahuje veškerou logiku. Komunikuje s modelem a zpracovaná data přeposílá pohledu. V Nette Frameworku se jedná o potomky odvozené od třídy Presenter.

4.1.4 Databáze

Jako SŘBD je zvolena MySQL databáze z důvodu bezplatné licence, rozšířenosti a snadné implementovatelnosti. V Nette se k databázi přistupuje pomocí NotORM, což je nástroj pro usnadnění práce s relační databází.

4.2 Návrh databáze

V lineárním zápisu je popsána struktura databáze v logickém pořadí. V následujícím soupisu tabulek z databáze jsou uvedeny tabulky JINEOBJEKTY a KONKRETNIPRVEK. Tyto tabulky zde fungují jako vzory pro **Rozhraní** (viz. kapitola 2.3) a **Prvky** projektu. Všechny mají podobnou strukturu a liší se pouze danými parametry, které jsou uvedeny jako PARAMETRY uvnitř závorek. Na tyto parametry mohou být navázány i nově vzniklé tabulky, pokud jsou třeba. Základní **Rozhraní** se nazývá **Vlastní** a jedná se o tabulku Hlavní. Přes ní se k objektu navážou jednotlivé **Prvky**. Ostatní **Rozhraní** jsou vytvářena podle vzoru JINEOBJEKTY. Naproti tomu podle vzoru KONKRETNIPRVEK se vytvářejí nové **Prvky** stránky jako jsou například: texty, obrázky, videa, mapy..., z kterých se skládá základní **Rozhraní**. Na tomto místě je také možné rozšíření o další nové **Prvky**.

4.2.1 Lineární zápis

Uzivatel (id, login, heslo, mail, jmeno, prijmeni, pohlavi, narozen, jeaktivovan)

Projekt (id, nazev, barva, **uzivatel_id**, **stranka_id**)

Stranka (id, nazev, **obsah_id**, **leve_id**, **prave_id**, **hlava_id**, **pata_id**, **projekt_id**)

Hlava (id, nazev, nadpis, barva, barva_text, obrazek, **projekt_id**)

Pata (id, nazev, text, barva, barva_text, **projekt_id**)

Leve (id, nazev, strom, barva, barva_aktivni, barva_text, barva_aktivni_text, barva_hlavni, **projekt_id**, **spoj_id**)

Prave (id, nazev, strom, barva, barva_aktivni, barva_text, barva_aktivni_text, barva_hlavni, **projekt_id**, **spoj_id**)

Horni (id, nazev, barva, barva_aktivni, barva_text, barva_aktivni_text, barva_hlavni, **projekt_id**, **spoj_id**)

Spoj (id)

Polozka (id, text, odkaz, poradi, **spoj_id**, **polozka_id**)

Obsah (id, nazev, titulek, barva, barva_text, barva_nadpis, **projekt_id**, **objekt_id**)

Objekt (id, typ, nazev, **uzivatel_id**)

JINEOBJEKTY (id, PARAMETRY, **objekt_id**)

Hlavni (id, **objekt_id**)

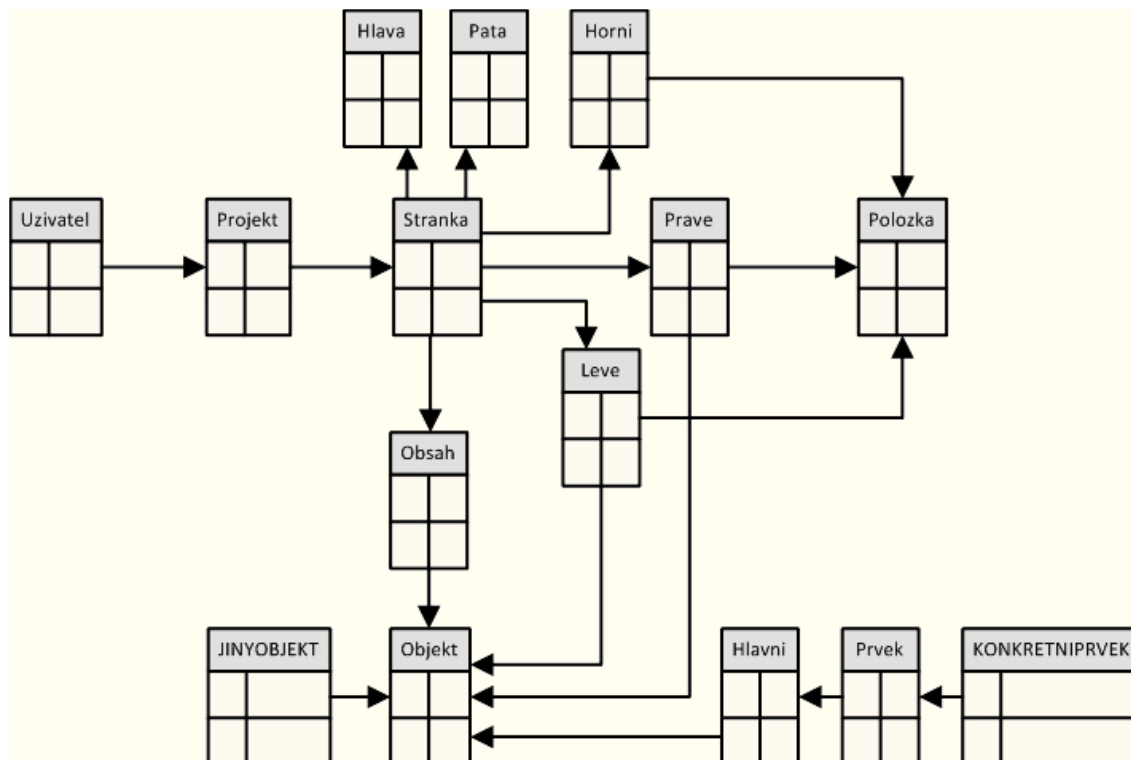
Prvek (id, sloupcu, poradi, **hlavni_id**)

KONKRETNIPRVEK (id, PARAMETRY, **prvek_id**)

Legenda: primární klíč, **cizí klíč**, povinný argument

4.2.2 Schéma databáze

Následuje schéma samotné databáze na obrázku č. 6.



Obrázek 6: Schéma databáze

4.3 Bezpečnost

Důležitou součástí každého webového projektu je i bezpečnost. O bezpečnost se stará samotný Nette Framework. Zde jsou uvedeny možné útoky a jejich ochrana:

- **Cross-Site Scripting (XSS)** - jedná se o metodu narušení webových stránek zneužívající neošetřené výstupy. Nette Framework všechny výstupy automaticky ošetřuje a tak se nemůže stát, že by se někde objevil neošetřený výstup.
- **Cross-Site Request Forgery (CSRF)** - útok, který přiměje uživatele navštívit stránku, která skrytě vykoná útok na aplikaci, kde je uživatel přihlášen. V Nette Frameworku se proti tomuto útoku dá bránit jednoduchým příkazem u formuláře:

```
$form->addProtection();
```

Výpis 3: Ochrana formulářů

Takto jsou v projektu chráněny všechny formuláře v administrační části webu.

- **URL attack, control codes, invalid UTF-8** - další typy útoků, které se snaží podstrčit webové aplikaci škodlivý kód. Nette Framework opět automaticky vše ošetří.
- **Session hijacking, session stealing, session fixation** - útoky spojené se správou session. Útočník zcizí a nebo podstrčí falešné session ID a získá tak přístup do webové aplikace, aniž by znal heslo. Ochrana spočívá ve správné konfiguraci serveru a PHP. Nette Framework vše nakonfiguruje automaticky.

Z předchozího výčtu možných útoků vyplývá, že o naprostou většinu bezpečnostních rizik se postará samotný Nette Framework. Zbývá ochrana hesel. Pro ochranu hesel je využito jednosměrné šifrování pomocí hašovací funkce sha512. O šifrování hesel se stará následující funkce:

```
public function calculateHash($password) {  
    return hash('sha512', "první" . $password . "druhá");  
}
```

Výpis 4: Šifrování hesel

Bezpečnost uložených hesel je navíc zvýšena tzv. solením [9]. Z důvodu obrany proti použití rainbow tables, což jsou tabulky dvojic, podle kterých se dá zjistit předchozí heslo z uložené šifry. Přidáním textu k již vytvořenému heslu totiž dojde k pozměnění výsledného zašifrovaného hesla a tím jsou rainbow tables pro zjištění původního nepoužitelné.

5 Popis implementace

Celý systém je navržen jako webová aplikace a implementován pomocí Nette Frameworku. Nette Framework je populární nástroj pro vytváření webových aplikací v PHP. Eliminuje bezpečnostní rizika, podporuje AJAX a znovupoužitelnost. Protože o Nette Frameworku zatím nevyšla žádná kniha byla k vývoji systému využita dokumentace a fórum z oficiálních webových stránek Nette Frameworku. Z webových technologií jsou pro implementaci systému využity následující: HTML, CSS, PHP, JavaScript, AJAX a JQuery.

Jako SŘBD je zvolena nej dostupnější relační databáze MySQL s využitím NotORM.

Jako vývojové prostředí je zvolen volně šířitelný univerzální editor PSPad a pro práci s databází je využit Adminer. Jedná se o nástroj pro správu databáze psaný v PHP, který oproti více známému phpMyAdminu umožňuje využívat triggerů a má mnoho dalších výhod.

Dále jsou uvedeny programátorské postupy, které jsou využity při programování systému a problémy, které se při vývoji objevily.

5.1 Vlastní ovládací prvky pro formuláře

V této části je popsána tvorba vlastních ovládacích prvků určených pro tento publikační systém. Jedná se o prvky pro zadávání data, barvy, odkazů...

5.1.1 Komponenta pro datum

Pro zadávání data narození je využita veřejně přístupná JavaScriptová komponenta od Jasona Moona. Tento skript je počestěn a je pozměněn jeho vzhled. Pro integraci do systému Nette je napsána nová komponenta pro formulář, nazvaná **DatePicker**. Základní vlastnosti jsou zděděny ze třídy TextInput, která už je v Nette zakomponována a představuje v HTML vstupní pole input. V nové komponentě DatePicker jsou přepsány téměř všechny funkce. Za zmínku stojí funkce SetValue pro nastavení hodnoty:

```
public function setValue($value) {
    if (preg_match('#(?:P<dd>\d{1,2})[.-]*P<mm>\d{1,2})([.-]*P<yyyy>\d{4})?$', $value, $matches)) {
        $dd = $matches['dd'];
        $mm = $matches['mm'];
        $yyyy = isset($matches['yyyy']) ? $matches['yyyy'] : date('Y');
        if (checkdate($mm, $dd, $yyyy)) {
            $value = "$yyyy-$mm-$dd";
        } else { $value = NULL; }
    }
    if ($value) {
        $this->value = new \Nette\DateTime($value);
        return $this;
    } else return NULL;
}
```

Výpis 5: Funkce SetValue pro DatePicker

Po implementaci komponenty DatePicker je použití data jednoduché. V Presenteru stačí přidat následující kód:

```
$form->addDatePicker('narozen', 'Datum_narození:', true, 'DD-MM-YYYY')
->addRule(Form::RANGE, 'Musíte_zadat_platné_datum_narození.', array(new DateTime('
-50000_days'), new DateTime('-365_days')));
```

Výpis 6: Vložení komponenty DatePicker do presenteru

Při vytváření se rovnou určí rozsah platnosti data narození. V tomto případě se jako platné datum bere interval, který začíná před 50000 dny, což je přibližně 136 let a jako konec intervalu se bere rok zpátky.

O vykreslení této komponenty se stará následující kód:

```
<div id="datumFormular1">
  {label narozen /}
  <div class="input">
    <script type="text/javascript">DateInput('narozen', true, 'DD-MM-YYYY')</script>
  </div>
</div>
```

Výpis 7: Vykreslení komponenty DatePicker

5.1.2 Vstup barvy

Pro výběr barvy je zvolena volně dostupná JSColor komponenta od Jana Odvarko.

K zprovoznění stačí uvést cesty k souborům do hlavní šablony projektu. Pro změnu běžného HTML inputu na JSColor input stačí danému inputu v presenteru nastavit CSS třídu **color**. Zde je ukázka kódu:

```
$form->addText('barva_text', 'Barva_textu:', 10, 10)->setDefaultValue('000000')
->getControlPrototype()->setClass('color');
```

Výpis 8: Nastavení JSColor inputu

Jedná se o vstupní pole pro zadání barvy. V této ukázce je nastavena výchozí barva na černou (000000).

5.1.3 Interní a externí odkaz

Způsob zadávání odkazů je vytvořen pomocí JavaScriptu. Jedná se o výběrovou nabídku (select box) a textový vstup (input). Výběrová nabídka nabízí 2 druhy hodnot na výběr:

1. **Externí odkaz** - textové pole se zpřístupní pro nakopírování odkazu na externí stránky
2. **Interní odkazy** - v nabídce se nachází seznam všech stránek daného projektu. Při výběru nějaké interní stránky se textové pole smaže a zneprístupní

Tento systém přidání odkazu je jednoduchý na používání. Zde následuje kód JavaScriptové funkce SelectLink, která je použita pro tyto odkazy:

```
function selectLink() {
    var pagesObj = document.getElementById("frmpolozkyForm-odkaz");
    if (pagesObj.options[pagesObj.selectedIndex].value == 0) {
        document.getElementById("frmpolozkyForm-odkaz2").disabled = false;
        document.getElementById("frmpolozkyForm-odkaz2").value = "http://";
    }
    else {
        document.getElementById("frmpolozkyForm-odkaz2").value = "";
        document.getElementById("frmpolozkyForm-odkaz2").disabled = true;
    }
};
```

Výpis 9: Funkce pro odkazy v JavaScriptu

5.1.4 Vstupní pole pro vkládání videí

Videa se vkládají pomocí odkazu ze serveru <http://www.youtube.com/>. O rozparsování odkazu do požadované podoby se stará funkce YoutubeOdkaz:

```
public function YoutubeOdkaz ($odkaz)
{
    $t2 = str_replace('watch?v=', 'v/', $odkaz);
    $t3 = $t2;
    if ($pos = strpos($t2, '&'))
        $t3 = substr($t2, 0, $pos);
    return $t3;
}
```

Výpis 10: Funkce pro převod odkazu videa v PHP

5.1.5 Vstupní pole pro vložení mapy

Mapy se vkládají podobně jako videa. S mapami se v systému pracuje pomocí Google Maps API, což je rozhraní umožňující vložení mapy na webovou stránku. K vložení mapy stačí zkopírovat odkaz mapy ze serveru <http://maps.google.com/>. Z odkazu je důležité získat údaje o souřadnicích daného místa. K rozparsování jsou použity následující 3 funkce:

```
public function mapa_pozice($odkaz)
{
    $t2 = $odkaz;
    $t3 = $t2;
    if ($pos = strpos($t2, 'll='))
        $t3 = substr($t2, $pos+3, -1);
    return $t3;
}
public function mapa_sirka($odkaz)
{
    $t2 = $this->mapa_pozice($odkaz);
    $t3 = $t2;
```

```

if ($pos = strpos($t2, ','))
    $t3 = substr($t2, 0, $pos);
return $t3;
}
public function mapa_vyska($odkaz)
{
    $t2 = $this->mapa_pozice($odkaz);
    $t3 = $t2;
    if ($pos = strpos($t2, ','))
        if ($pos2 = strpos($t2, '&'))
            $t3 = substr($t2, $pos+1, $pos2-$pos-1);
    return $t3;
}

```

Výpis 11: Funkce pro převod odkazu mapy v PHP

5.2 Stromová nabídka

Pro levou a pravou nabídku jsou přidány možnosti zaškrťovací volby, zda nabídka bude stromová a nebo normální. Z pohledu implementace je na stromové nabídce zajímavé rekurzivní vykreslení.

Před samotnou ukázkou vykreslení je vysvětleno, jak funguje přidávání položek do postranní nabídky. V databázi má tabulka **Polozka** atributy **spoj_id** a **polozka_id**. **Spoj_id** slouží k napojení na tabulku **Spoj**, která slouží k napojení na všechny 3 nabídky, které jsou v systému (horní, levá a pravá). **Polozka_id** slouží k napojení na předky. Přes tento atribut probíhá samotná rekurze. U horní nabídky má **polozka_id** vždy hodnotu **NULL**.

Zde je uvedena ukáзка vykreslení levé stromové nabídky:

```

{ if $kategorie->leve->strom == 1 }
<div id="lstrom">
<ul id="koren" class="menu">
    {block #menu_strom}
    {foreach $lpolozky as $lpolozka}
    { if $model->getPolozka()->where('polozka_id', $lpolozka->id)->order('poradi')->
        count('*')}
    <li>
        { if is_numeric($lpolozka->odkaz)}
        <a href='javascript:void(0);' potomekid='c_{ $lpolozka->id}' class='cat_close_
            category'>&nbsp;</a><a href='{plink_Nahled:.$lpolozka->odkaz}'>{
                $lpolozka->text}</a>
        {else}
        <a href='javascript:void(0);' potomekid='c_{ $lpolozka->id}' class='cat_close_
            category'>&nbsp;</a><a href='{ $lpolozka->odkaz}'>{ $lpolozka->text}</
            a>
        {/if}
    </li>
    <ul id='c_{ $lpolozka->id}'>
        {include #menu_strom, 'lpolozky' => $model->getPolozka()->where('polozka_id',
            $lpolozka->id)->order('poradi')}
    </ul>
    {else}

```

Pro samotnou rekurzi jsou v Nette Frameworku použity tzv. Bloky, které slouží k označení části šablony a k opětovnému znovuvložení. Pomocí bloku **menu_strom** probíhá samotná rekurze. Jako parametry se do bloku předávají potomci dané položky nabídky seřazení podle pořadí. Jedno zavolání bloku **menu_strom** značí vykreslení jedné úrovně stromu dané větve.

Návštěvní kniha je druhým typem **Rozhraní**. Pro stránkování příspěvků je použit stránkovač z Nette Frameworku, který se nazývá **Paginator** (je použit také v administrační části). Využitá je zde rekurze založená na stejném principu jako stromová nabídka výše. Atribut pro rekurzi **reakce_id** je obdobou výše zmíněného atributu **polozka_id** u nabídky.

[illegible]

```

    { if $model->getKniha()->where('objekt_id', $kategorie->obsah->objekt->id)->
      fetch()->reakce == 1 }
    {include #prispevek, 'prispevky' => $model->getKnihaPrispevky()->where('
      reakce_id', $prispevek->id)} // rekurzivní volání
    {/if}
  </div>
</div>
{/foreach} // konec cyklu
{/block} // konec bloku
{control vp} // vykreslení stránkování

```

Výpis 13: Rekurzivní vykreslení návštěvní knihy

5.3.1 Reakce

Pokud chce uživatel reagovat na nějaký příspěvek, musí kliknout na klikací odkaz **reagovat**, který se nachází vpravo nahoře u příspěvku. Kliknutí na tento odkaz zavolá JavaScriptovou funkci, která do vstupního pole návštěvní knihy **Reakce** vloží název a id příspěvku. Pro zrušení reakce stačí kliknout na pole **Reakce** a dojde k jeho vymazání. Zde je kód funkcí **reakce_smaz** a **reaguj**:

```

function reakce_smaz() {
  document.getElementById("frmpridatForm-reakce").readonly = false;
  document.getElementById("frmpridatForm-reakce").value = null;
  document.getElementById("frmpridatForm-reakce").readonly = true;
  document.getElementById("frmpridatForm-reakce_id").value = null;
};
function reaguj(cislo, nadpis) {
  document.getElementById("frmpridatForm-reakce").readonly = false;
  document.getElementById("frmpridatForm-reakce").value = nadpis;
  document.getElementById("frmpridatForm-reakce").readonly = true;
  document.getElementById("frmpridatForm-reakce_id").value = cislo;
};

```

Výpis 14: Funkce pro reakce v návštěvní knize

5.4 Použití triggerů a transakcí pro mazání

Pro mazání jednotlivých záznamů v tabulkách jsou použity většinou triggerů na straně databázové vrstvy a transakce na straně aplikační vrstvy.

Pro smazání **Prvku** stačí pouze 1 řádek kódu na aplikační vrstvě:

```

$this->model->getPrvek()->where('array('id' => $projektId))->delete();

```

Výpis 15: Příkaz pro smazání Prvku

O smazání vázaných záznamů se postará trigger na straně databázového serveru, který se spustí před smazáním samotného **Prvku**:

```

CREATE TRIGGER 'prvek_bd' BEFORE DELETE ON 'prvek' FOR EACH ROW
BEGIN

```

```
DELETE FROM mezera WHERE mezera.prvek_id = OLD.id;
DELETE FROM odstavec WHERE odstavec.prvek_id = OLD.id;
DELETE FROM obrazek WHERE obrazek.prvek_id = OLD.id;
DELETE FROM video WHERE video.prvek_id = OLD.id;
END;
```

Výpis 16: Trigger Prvku před smazáním

Při smazání **Rozhraní Vlastní** se volají 2 triggery. První trigger se volá před smazáním základního **Rozhraní**:

```
CREATE TRIGGER 'hlavni_bd' BEFORE DELETE ON 'hlavni' FOR EACH ROW
DELETE FROM prvek WHERE prvek.hlavni_id = OLD.id;
```

Výpis 17: Trigger základního Rozhraní před smazáním

Zavoláním tohoto triggeru se smaže **Prvek**, který před vlastním smazáním vyvolá trigger **prvek_bd** zobrazený výše. Tím se postará o smazání vázaných tabulek na **Prvek**.

Po samotném smazání základního **Rozhraní** se volá druhý trigger:

```
CREATE TRIGGER 'hlavni_ad' AFTER DELETE ON 'hlavni' FOR EACH ROW
DELETE FROM objekt WHERE objekt.id= OLD.objekt_id;
```

Výpis 18: Trigger základního Rozhraní po smazání

Tento trigger se postará o smazání záznamů z tabulky **Objekt**, která slouží k navázání **Rozhraní** na obsah nebo postranní nabídku. Tabulka **Objekt** má definován trigger před smazáním, který nastaví kvůli zrušení vazby obsahu a nabídek vázaný atribut na hodnotu **NULL**.

```
CREATE TRIGGER 'objekt_bd' BEFORE DELETE ON 'objekt' FOR EACH ROW
BEGIN
UPDATE obsah SET objekt_id = NULL WHERE obsah.objekt_id = OLD.id;
UPDATE leve SET objekt_id = NULL WHERE leve.objekt_id = OLD.id;
UPDATE prave SET objekt_id = NULL WHERE prave.objekt_id = OLD.id;
END;
```

Výpis 19: Trigger Objektu

5.4.1 Rekurzivní mazání pomocí transakcí a triggerů

Problematické je rekurzivní mazání záznamů z tabulky. Trigger totiž nesmí jakkoliv upravovat položky Tabulky z které je volán. Z toho důvodu je pro rekurzivní mazání vytvořena pomocná tabulka.

Rekurzivní mazání je vysvětleno na smazání celé levé nabídky. U pravé nabídky to funguje obdobně. Smazáním celé nabídky se zavolá následující transakce, která je okomentována:

```
$this->model->getDb()->beginTransaction(); // začátek transakce
// Nejdříve dojde ke zrušení vazby s tabulkou Stranka
$this->model->getStranka()->where(array('leve_id' => $projektId))->update(array(
'leve_id' => NULL
```

```

));
// nalezení ID tabulky Spoj
$spojid = $this->model->getLeve()->where(array('id' => $projektId))->fetch()->spoj->id;
// smazání levé nabídky
$this->model->getLeve()->where(array('id' => $projektId))->delete();
// výběr položek menu
$polozky = $this->model->getPolozka()->where(array('spoj_id' => $spojid));
foreach ($polozky as $polozka){ // průchod položkami daného levého menu
    // zrušení vazby položky s levým menu
    $this->model->getPolozka()->where(array('id' => $polozka->id))->update(array(
        'spoj_id' => NULL
    ));
}
// smazání záznamu z tabulky Spoj
$this->model->getSpoj()->where(array('id' => $spojid))->delete();
// smazání pomocné tabulky Polozka_trigger
$this->model->getPolozkaTrigger()->delete();
$this->model->getDb()->commit(); // konec transakce

```

Výpis 20: Transakce pro smazání levé nabídky

Samotná rekurze probíhá na posledních 2 řádcích. Po smazání záznamu z tabulky **Spoj** se zavolá následující trigger:

```

CREATE TRIGGER 'spoj_ad' AFTER DELETE ON 'spoj' FOR EACH ROW
BEGIN
    DELETE FROM polozka WHERE polozka.spoj_id = OLD.id;
    INSERT INTO polozka_trigger (SELECT id FROM polozka WHERE polozka.spoj_id IS NULL AND
        polozka.polozka_id IS NULL);
END;

```

Výpis 21: Trigger pro Spoj po smazání

Tento trigger smaže položky, na které se vázal mazaný záznam tabulky **Spoj**. Poté vloží do pomocné tabulky **Polozka_trigger** všechny potomky smazaných položek. V předchozí transakci se provede poslední příkaz a všechny položky, které se před chvílí do databáze vložily se nyní smažou. Po smazání těchto položek z pomocné tabulky se zavolá následující trigger:

```

CREATE TRIGGER 'polozka_trigger_ad' AFTER DELETE ON 'polozka_trigger' FOR EACH ROW
BEGIN
    UPDATE polozka SET polozka.polozka_id = NULL WHERE (polozka.spoj_id IS NULL) AND (polozka
        .polozka_id = OLD.id);
    DELETE FROM polozka WHERE polozka.id = OLD.id;
END;

```

Výpis 22: Trigger pro pomocné tabulky

Tento trigger zruší vazbu s potomkem a hned poté smaže položku s daným id.

Po těchto triggerech došlo tedy ke smazání jedné úrovně stromu. Položka, která má smazanou vazbu s potomkem má nyní obě vazby nastaveny na **NULL**. Díky tomu je volně v databázi a do systému nebude nijak zasahovat. Při dalším mazání

jakékoliv nabídky se opět zavolá trigger **spoj_ad** zobrazený výše a smaže se další úroveň useknutého stromu.

Tímto pseudorekurzivním způsobem je šetřen výkon databáze pro smazání jen 1 větve. Mazání dalších větví se provádí postupem času při volání mazání nabídek.

5.5 Vzhled systému

Zde následují ukázky vzhledu systému. Stránky jsou v prohlížeči vždy vycentrovány na střed. Na ukázkách je odříznuta prázdná levá část obrázků z administrační části, tak aby byl v náhledu vidět lepší detail aplikace.

5.5.1 Hlavní strana - registrace

Vzhled administrační části je zobrazen na obrázku č. 7.

Obrázek 7: Hlavní strana - registrace

V pravé horní části stránky můžeme vidět odkaz **Přihlásit se**, který slouží k přihlášení uživatele. Uprostřed se nachází registrační formulář. Vlevo se nachází jen ta část levé nabídky, která je určena pro nepřihlášeného uživatele.

5.5.2 Náhled administrace

Náhled administrace je zobrazen na obrázku č. 8.

The screenshot displays the 'Z-WARE - levé menu' administration page. The interface includes a sidebar on the left with navigation links: Úvod, Tutoriál, Dokumentace, Projekty (with sub-items TEST and Z-WARE), Hlavička, Horní menu, Levé menu, Obsah, Pravé menu, and Pata. Below this are sections for Rozhraní (Vlastní, Návštěvní kniha) and Změna hesla. A 'Nový projekt' form is also present.

The main content area is titled 'Z-WARE - levé menu' and contains a form to 'Přidat levé menu'. This form includes fields for 'Název' (NOVÉ MENU), 'Stromové menu' (checked), 'Barva pozadí' (FFFFFF), 'Barva textu' (000000), 'Barva pozadí 2' (FFFFFF), 'Barva textu 2' (000000), and 'Obsahuje' (Není -). A 'Vytvořit levé menu' button is at the bottom.

Below the form is a table listing existing menu items:

ID	Název	Barva pozadí	Barva textu	Barva pozadí 2	Barva textu 2	Stromové menu	Obsahuje
19	Levé menu	00518F	E0F8FF	00518F	FFFFFF	ano	není

Below the table are links for 'Spravovat položky', 'editovat', and 'smazat'. Below the table is a section for 'Editace položky číslo 19' with a form to edit the menu item. The form includes fields for 'Název (19):', 'Stromové menu' (checked), 'Barva pozadí' (00518F), 'Barva textu' (E0F8FF), 'Barva pozadí 2' (00518F), 'Barva textu 2' (FFFFFF), and 'Obsahuje' (- není -). An 'Editovat' button is at the bottom.

The footer of the page shows copyright information: 'Copyright © Záškodný Marek 2012 (ZAS0006)' and a debug bar with performance metrics.

Obrázek 8: Náhled administrace

V této ukázce můžeme vidět administraci levé nabídky. Vpravo nahoře je odkaz pro odhlášení **Odhlásit se**. Vpravo dole se nachází Debugger Bar, což je plovoucí informační panel od Nette.

Na levé části se nachází nabídka, která je oproti předcházející ukázce větší. Obsahuje už položky, které jsou dostupné jen přihlášenému uživateli. Nově se tu objevila část s vytvořenými projekty, část s **Rozhraním**, změna hesla a formulář v nabídce pro vytvoření nového projektu.

Obsah stránky začíná nadpisem, kde je uvedeno, že se jedná o správu projektu Z-WARE a že jsme konkrétně v části kde se spravuje levá nabídka. Poté následuje formulář pro tvorbu levé nabídky, přehled levé nabídky s možností správy položek a nejnižší je otevřena editace konkrétní položky.

6 Popis jednotlivých částí publikačního systému

V této části bakalářské práce se už dostáváme podrobně k popisu jednotlivých částí celého publikačního systému od registrace až po jednotlivé funkce systému.

6.1 Registrace a přihlášení

V registraci se vyplňují následující údaje:

- Uživatelské jméno
- Heslo
- E-mail
- Pohlaví
- Jméno
- Příjmení
- Datum narození

Z toho nepovinné položky jsou jméno, příjmení a datum narození. Po vyplnění registrace je nutné aktivovat si účet přes e-mail na který ihned po registraci dorazí zakódovaný login pomocí této funkce:

```
public function zakoduj($retezec) {
    return hash('sha512', "sul" . $retezec . "pepr");
}
```

Výpis 23: Šifrování kontrolního kódu

Po potvrzení kontrolního odkazu z e-mailu se aktivuje účet a je možné se přihlásit do administrační části systému.

6.2 Správa projektu

Nový projekt je možno vytvořit v levé nabídce dole. Po pojmenování projektu a potvrzení se vytvoří záznamy v 8 tabulkách databáze. Takže ihned po vytvoření projektu se objeví základní webová prezentace, která obsahuje 1 obrázek a 1 odstavec textu pro ukázkou.

V levé nabídce se navíc zpřístupní nová sekce s názvem daného projektu, kde může uživatel samostatně spravovat hlavičku, horní nabídku, levou nabídku, obsah, pravou nabídku a patičku nového projektu. Levá nabídka obsahuje také položku **Projekty**, kde je možné spravovat všechny projekty. V editaci projektu je možné změnit název, barvu pozadí a hlavní stranu projektu.

6.3 Správa stránek

Přehled stránek projektu se objeví po kliknutí na název projektu v levé nabídce administrace. Pro vytvoření nové webové stránky stačí zadat název stránky a vybrat ze seznamů z jakých částí se daná stránka bude skládat. Každá stránka se skládá z následujících částí:

- Hlavička
- Horní nabídka
- Levá nabídka
- Obsah
- Pravá nabídka
- Patička

Jedinou povinnou částí je Obsah. Ostatní jsou volitelné. K vytvoření stránky tedy stačí zadat pouze název a obsah.

6.4 Hlavní části webové stránky

Ke všem částem se jednotně přistupuje přes levou rozbalovací nabídku po najetí na název projektu. Tato část se týká všech součástí v pořadí od jednoduchých ke složitějším částem. Všechny části webové stránky se spravují samostatně.

Začínáme hlavičkou a patičkou. Jedná se o nejjednodušší části webových stránek. Jejich správa je jednoduchá a intuitivní, níže je vypsáno, co tyto části obsahují.

6.4.1 Patička

Pro vytvoření patičky je třeba zadat následující údaje:

- Název
- Nadpis
- Barva pozadí
- Barva textu
- Obrázek

6.4.2 Hlavička

Pro vytvoření hlavičky je třeba zadat následující údaje:

- Název
- Text
- Barva pozadí
- Barva textu

Další částí jsou nabídky. Nabídky se v projektu dělí na horní, levé a pravé. Dále se nabídky logicky dělí na horní a postranní (levé a pravé). To z toho důvodu, že postranní nabídky jsou obě totožné a založeny na fungování horní nabídky. Liší se od ní jen tím, že umožňují zobrazovat nabídku ve stromové struktuře.

6.4.3 Horní nabídka

Údaje pro vytvoření horní nabídky:

- **Název**
- **Barva pozadí**
- **Barva textu**
- **Barva pozadí 2**
- **Barva textu 2**

Pod formulářem pro vytvoření nabídky se nachází přehled všech horních nabídek.

U každé nabídky je klikací odkaz **Spravovat položky**. Kliknutím na tento odkaz se přesunete k vlastní tvorbě jednotlivých položek nabídky. Pro vytvoření položky nabídky stačí vyplnit text a odkaz. U odkazu je možno se odkazovat na lokální již vytvořené stránky a nebo na jakékoliv externí webové stránky pomocí odkazu. Jednotlivé položky nabídky můžete editovat a jednoduše měnit jejich pořadí pomocí klikacích šipek nahoru a dolů.

Řazení je založeno na podobném principu jako v Joomla!. Náhled řazení v Joomla se nachází v první kapitole na obrázku č. 3.

6.4.4 Levá a pravá nabídka

Levá a pravá nabídka funguje stejně jako horní nabídka. Navíc je zde možnost zaškrtnout si pomocí checkboxu, zda daná nabídka bude ve stromové a nebo normální (stejně jako horní) struktuře. Při výběru stromové nabídky je navíc u jednotlivých položek nabídky klikací odkaz **Spravovat podkategorie**. Po kliknutí na tento odkaz se otevře nová stránka, kde je možno vytvářet nové položky stejně jako na stránce předchozí. Tyto položky se vážou na předcházející položky a tím se vytváří stromová nabídka. Počet zanoření není omezen. V editaci postranní nabídky je možno po vytvoření změnit nabídku zpět na normální. Při výběru normální nabídky se vnořené položky skryjí a nabídka se spravuje stejně jako horní nabídka.

Dále je možné pod nabídku vložit jakékoliv **Rozhraní**. Funguje to stejně jako údaj Obsahuje v Obsahu níže.

6.4.5 Obsah

Pro vytvoření obsahu je třeba zadat následující údaje:

- **Název**
- **Titulek** - vloží se do hlavičky mezi tagy title, má velký význam pro SEO stránek
- **Barva pozadí**
- **Barva textu**
- **Barva nadpisu**
- **Barva nabídky**
- **Obsahuje**

Do části Obsahuje patří samotný obsah stránek (texty, obrázky, videa, návštěvní kniha), který se vytváří níže.

6.5 Správa Rozhraní

Rozhraní slouží k napojení na obsah a postranní nabídky stránek. Určuje jejich obsah. Je to prostředek určený pro napojení různých objektů.

6.5.1 Základní rozhraní - Vlastní

Základním **Rozhraním** aplikace je objekt zvaný **Vlastní**. Nazývá se **Vlastní** z důvodu, že jednotlivé **Prvky** tohoto **Rozhraní** si definuje samotný uživatel. Nachází se v levé nabídce pod projekty. K vytvoření **Rozhraní Vlastní** stačí zadat pouze název. Po vytvoření se objeví v seznamu pod formulářem.

Rozhraní Vlastní jak už vyplývá z předchozího odstavce se skládá z **Prvků**. Napojení **Prvků** na **Rozhraní Vlastní** je podobné jako napojení **Rozhraní** na obsah stránky. Podobně jako se v menu spravují položky je možné spravovat **Prvky Rozhraní Vlastní** pomocí klikacího odkazu **Spravovat prvky**. **Prvky** stránek s vyplňovanými údaji jsou následující:

- **Odstavec** - nadpis a text
- **Obrázek** - obrázek a popis
- **Mezera** - odděluje sloupce pomocí CSS vlastnosti clear:both [7], která zajistí zakázání obtékání plovoucích objektů (v tomto případě sloupců) po obou stranách.
- **Video** - odkaz na video ze serveru <http://www.youtube.com/>, zde je možné upravit funkci pro rozšíření o více video portálů.
- **Mapa** - odkaz na mapu ze serveru <http://maps.google.cz/> a přiblížení mapy.

U všech **Prvků** se vyplňuje šířka a druh **Prvku** z předchozího seznamu. Šířka má 6 možných hodnot: celá šířka, 1/2 šířky, 1/3 šířky, 1/4 šířky, 2/3 šířky a 3/4 šířky. Jedná se o šířku obsahu. Všechny prvky je tedy možné poskládat až do 4 sloupců a je možné pomocí klikacích šipek nahoru a dolů měnit jejich pořadí. Po vytvoření **Prvku** je třeba daný **Prvek** editovat a přiřadit mu obsah podle daného typu **Prvku** ze seznamu.

6.5.2 Návštěvní kniha

Návštěvní kniha je další druh **Rozhraní**. Pro vytvoření nové návštěvní knihy je potřeba vyplnit tyto údaje:

- **Název**
- **Barva pozadí**
- **Barva textu**
- **Barva příspěvku**
- **Příspěvků na stranu** - určuje stránkování návštěvní knihy
- **Reakce v knize**

Pokud je zaškrtnuta volba reakce v knize, tak bude v návštěvní knize možné navíc reagovat na jakýkoliv příspěvek. Při reakci se odpověď vypíše do vnořeného rámečku pod příspěvek na který se reaguje. Jedná se o podobný princip jako u stromové postranní nabídky. Reakce v knize je možné kdykoliv za běhu aplikace vypnout a opětovně zapnout. Při vypnutí stejně jako u nabídky dojde pouze k skrytí reagujících příspěvků.

V případě použití návštěvní knihy v postranních nabídkách jsou příspěvky zobrazeny bez reakcí a bez stránkování. Dá se pouze navolit kolik posledních příspěvků se má zobrazit. Návštěvní kniha tedy pro nabídky funguje v podstatě jako chat.

7 Případová studie ukázkové prezentace na reálném projektu

Jako reálný projekt jsou zvoleny webové stránky firmy **Z-WARE**. Jedná se o firmu zabývající se výrobou a instalací identifikačních systémů. Stránky jsou v bakalářské práci použity se souhlasem majitele stránek pana Ing. Vladimíra Zavřela. Tato webová prezentace je zvolena pro ukázkou funkčnosti publikačního systému na reálném projektu. Obsahuje hlavní části, které by měla mít webová prezentace (hlavičku, nabídky, obsah a patičku). Dalším důvodem proč je tato firma zvolena pro případovou studii je, že celé webové stránky jsou obsáhlé, přehledné a mají celkem jednoduchý grafický design.

7.1 Porovnání vzhledu

Nyní následuje porovnání vzhledu originální prezentace s prezentací vytvořenou v publikačním systému.

Na obrázku můžeme vidět jednu ze stránek firmy **Z-WARE** v originální a nově vytvořené podobě.

7.1.1 Originální náhled

Vzhled originální prezentace je zobrazen na obrázku č. 9.

7.1.2 Vlastní náhled

Vzhled vlastní prezentace vytvořené podle původní je zobrazen na obrázku č. 10.

Tato výsledná prezentace, byla vytvořena pomocí tohoto publikačního systému. Jak můžeme vidět, tak jsou si stránky velmi podobné a jejich vytvoření zvládne i běžný uživatel.



Obrázek 9: Ukázka původní prezentace

Z-WARE

[Aktuality](#)
[Poptávkový formulář](#)
[Dotazy](#)
[Ke stažení](#)
[Kontakty](#)

▶

Docházkové systémy

■

Biometrická identifikace

▶

Stravovací systémy

▶

Přístupové systémy

▶

Domovní systémy

■

Ovládání kopírek

▶

Identifikační média

■

Systém Bakaláři

■

Systém SAS

■

Isic port

■

Zakázkové aplikace

■

Obchodní spolupráce

▶

Fotogalerie

▶

Ukázky z programů

Z-WARE

Naše firma se zabývá výrobou a instalací identifikačních systémů. Veškerý hardware i software je produktem naší firmy, a proto jsme schopni vyjit' maximálně vstříc zákaznickým potřebám.

DOCHÁZKOVÉ SYSTÉMY
přesný záznam pracovní doby, výpočet odpracovaných hodin a jednotlivých docházkových kont pro mzdy

IDENTIFIKAČNÍ MÉDIA
jednoznačná identifikace osob, výběr z několika typů, možnost použití jednoho média na všechny naše identifikační systémy

STRAVOVACÍ SYSTÉMY
objednávání a výdej stravy, finanční vyúčtování, skladové hospodářství, internetové objednávky, provoz bufetů

PŘÍSTUPOVÉ SYSTÉMY
přesná identifikace osob, libovolně nastavitelné časové zóny, zpětné sledování přístupů v software

OVLÁDÁNÍ KOPÍREK
pomocí čipů kontrola kopií kreditním systémem, stálý přehled o počtu zbývajících financí, včetně jednoduchého software pro dobíjení

Několik málo řádků o historii firmy Z-WARE. Naše firma vznikla v květnu 1969. V prvních letech jsme se zabývali především vývojem programového vybavení (simulátory technologických procesů, trenažéry pro jaderné i konvenční elektrárny, univerzální grafický a databázový editor a jiné podobné projekty).

Od roku 1996 se intenzivně zabýváme identifikačními systémy. Nejprve se jednalo o systém pro strážní službu, postupně jsme přešli k docházkovým systémům, systémům pro automatizaci stravování (objednávky, výdej, bezobjednávkový systém) a k přístupovým systémům, k ovládání kopírek, výrobě čipů, atd.

Tyto systémy jsou hlavní náplní naší firmy i v současnosti. Máme u všech identifikačních systémů více druhů software a hardware. Jeho vlastní nasazení závisí na požadavcích, finančních možnostech zákazníka. Všechny naše systémy jsou stavebnicové a dají se opět postupně rozšiřovat dle požadavků.

Hlavní střediska firmy jsou Jihlava a Brno. Další servisní místa jsou v Praze, Ostravě, Břeclavi, Lanškrouně a Ostravě. Stále rozšiřujeme síť spolupracujících firem. Disponujeme vývojovou kapacitou vlastní i dalších trvale spolupracujících firem. Nejdůležitější elektronické komponenty systému nejen vyvíjíme, ale i vyrábíme. Systémy projektujeme, montujeme, provádíme potřebnou údržbu a servis, popřípadě opravy po celé ČR.

Aktuálně provozujeme více než 700 identifikačních systémů po celé České republice. Dodáváme systémy pro malé i velké zákazníky. Největší provozované systémy mají více než 10 000 uživatelů. Průběžně nabízíme zákazníkům další nové výrobky, služby a zákaznické aplikace.

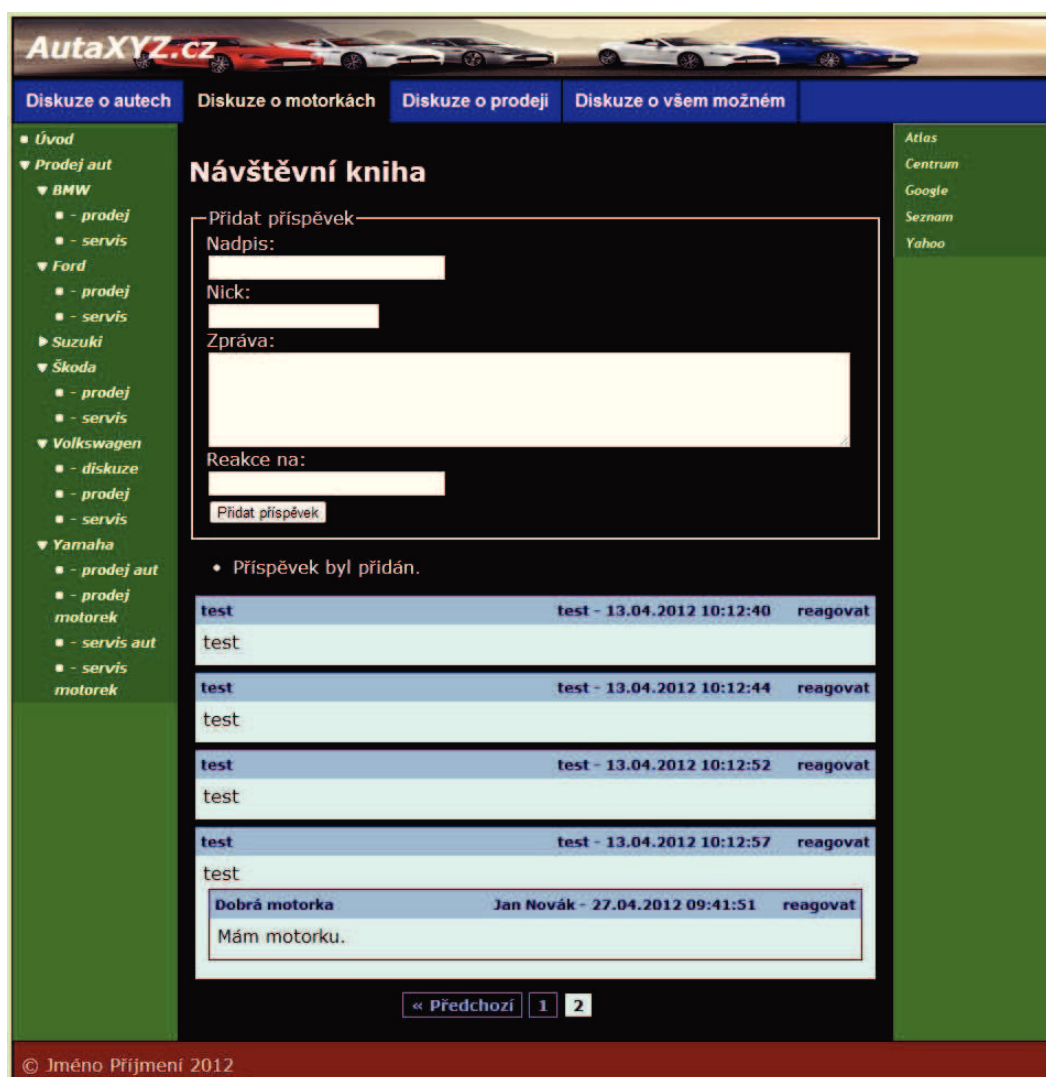
Validní XHTML 1.0 a CSS. Design © Blue Beret 2006–2008

Obrázek 10: Ukázka vlastní prezentace

7.2 Vlastní rozšířená prezentace

Na reálném projektu nebylo možné ukázat web se všemi funkcemi. Z toho důvodu následuje ukázka prezentace fiktivní firmy **AutaXYZ.cz**. Firma je vytvořena pouze pro ukázkou. V této ukázce web obsahuje všechny možné části - hlavičku, horní nabídku, levou nabídku, obsah, pravou nabídku a patičku. Levé nabídku je zobrazena ve stromové struktuře s dvojím zanořením položek. V obsahu stránky se nachází návštěvní kniha s povolenými reakcemi.

Vzhled rozšířené prezentace je zobrazen na obrázku č. 11.



Obrázek 11: Ukázka rozšířené prezentace - návštěvní kniha

Na této rozšířené webové prezentaci je vidět, že lze systém nasadit i na rozsáhlejší webové projekty než v předchozí ukázce. Horní nabídka slouží pro diskuze na různá

témata. V levé nabídce se nachází úvodní strana a dále značky aut u kterých je ještě podkategorie na prodej a nebo servis. V pravé nabídce se nachází externí odkazy na jiné webové stránky. Web také obsahuje klikací hlavičku a patičku s informací o tvůrci. V obsahu stránky se nachází již zmíněná návštěvní kniha s neomezeným počtem zanořených reakcí. Příspěvky v knize se stránkují po 10. Po návštěvní knize ještě následuje ukázka úvodní stránky na obrázku č. 12.



Obrázek 12: Ukázka rozšířené prezentace - úvodní strana

Na úvodní stránce je vidět způsob práce se sloupci, obrázky a videem. V ukázce je vidět, že se pod levou nabídkou nachází mapa a pod pravou nabídkou obrázek. Do postranních nabídek je možné vkládat stejné prvky jako do hlavního obsahu stránek.

7.3 Zhodnocení ukázkových prezentací

Na první části jsme si ukázali, že je možné v tomto publikačním systému vytvořit webovou prezentaci podle reálné webové prezentace firmy za velmi krátkou dobu bez znalosti programování.

Poté následovala ukázka rozšířeného webového projektu o další části a také o stromovou nabídku, která umožňuje lepší navigaci na webových stránkách a tvorbu rozsáhlejších webových stránek. V druhé ukázce se rovněž nachází návštěvní kniha s možností reagování na příspěvky a řazení obsahu do sloupců.

Výhody:

- Správa jednotlivých částí stránek samostatně.
- Rychlá a přehledná správa nabídek.
- Jednoduchá tvorba hlavičky a patičky.
- Samostatné nastavení vzhledu pro stránky bez ohledu na obsah.
- Integrace videa a mapy do obsahu i do postranních nabídek.

Nevýhody:

- Nedokonalý grafický design.
- Pevná šířka stránky.
- Omezení možností nastavení u hlavičky, horní nabídky a patičky.

8 Závěr

Cílem bakalářské práce bylo zachytit potřeby prezentace na internetu pro malé a střední webové projekty. A získané znalosti použít k návrhu a tvorbě vlastního publikačního systému, který by usnadnil vývoj webových prezentací. Výsledkem bakalářské práce je funkční publikační systém implementovaný v Nette Frameworku.

Oblast publikačních systémů je velmi obsáhlá a není možné, aby systém obsahoval všechna možná rozšíření spojená s CMS. Důraz je tedy kladen na ty části webových prezentací, které jsou nejdůležitější pro běžné firmy bez webových stránek.

Pro ukázkou funkčnosti celého systému je vybrána jako vzor reálná firma **Z-WARE**. Je vidět, že webová prezentace vytvořená v tomto publikačním systému je velmi podobná té původní a splňuje požadavky na webovou prezentaci pro malé a střední firmy.

Z pohledu dalšího vývoje je projekt jednoduše rozšiřitelný o nové funkce. Výsledný publikační systém je založen na architektuře, která je snadno rozšiřitelná o nové **Rozhraní a Prvky**. Dále by neměl být v budoucím vývoji problém výsledný projekt při exportu převést i na jiný typ databáze (např.: Oracle, PostgreSQL, MSSQL...). Tím by se ještě více zvýšila univerzálnost projektu.

Hlavní základ aplikace je nyní připraven k nasazení do skutečného provozu, kde je možné ověřit provozuschopnost při větší zátěži.

Jedním z nedostatků celého systému je grafické zpracování. Pokud se bude systém nadále vyvíjet, tak by bylo vhodné najmout grafika, který by vytvořil více šablon pro vzhled s možností stažení takového vzhledu, který by danému uživateli vyhovoval.

9 Literatura

- [1] BusinessInfo.cz: Oficiální portál pro podnikání a export [online]. CzechTrae, 1997-2011 [cit. 2012-04-20]. Dostupné z: <http://www.businessinfo.cz/cz/clanek/internet-pomaha-exportovat/vyznam-internetu-online-sluzby-exporteri/1001962/63591/>
- [2] Země internetová: Jak internet mění českou ekonomiku. ČERVENKA, Ladislav, Martin KÁLOVEC a Ondřej ŠABATA. Země internetová [online]. 2011 [cit. 2012-04-25]. Dostupné z: <http://www.zemeinternetova.cz/pdf/Report-zemeinternetova-8Mar2011-final.pdf>
- [3] LECKY-THOMPSON, Ed a Steven D NOWICKI. PHP 6: programujeme profesionálně. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně. ISBN 978-80-251-3127-5.
- [4] MCNULTY, Scott. WordPress: efektivní publikování na webu. Vyd. 1. Brno: Zoner Press, 2009, 256 s. ISBN 978-80-7413-042-7.
- [5] POLZER, Jan. Drupal: podrobný průvodce tvorbou a správou webů. Vyd. 1. Brno: Computer Press, 2008, 262 s. ISBN 978-80-251-1946-4.
- [6] KUBÍČEK, Michal. Velký průvodce SEO: jak dosáhnout nejlepších pozic ve vyhledávačích. Vyd. 1. Brno: Computer Press, 2008, 318 s. ISBN 978-80-251-2195-5.
- [7] HAUSER, Marianne. HTML a CSS: velká kniha řešení. Vyd. 1. Brno: Computer Press, 2006, 912 s. ISBN 80-251-1117-2.
- [8] Nette Framework: Rychlý a pohodlný vývoj webových aplikací v PHP — Nette Framework [online]. Nette Foundation, 2008, 2012 [cit. 2012-04-20]. Dostupné z: <http://nette.org/>
- [9] PHP Guru [online]. Jan Tichý, 2012 [cit. 2012-04-20]. Dostupné z: <http://www.phpguru.cz/clanky/soleni-hesel>

10 Přílohy

1. **Tutoriál** - je dostupný na adrese <http://zaskodny.cz/bakalarka/>, rovněž je uložen na CD pod názvem **tutorial.html**
2. **CD** - obsahuje bakalářskou práci, zdrojové kódy publikačního systému a výstup databáze
3. **Ukázkové webové prezentace** - uloženy v databázi na CD a ve funkční podobě se nacházejí na již zmíněné adrese <http://zaskodny.cz/bakalarka/> po přihlášení pod uživatelským jménem **test** a heslem **test123**. Vyexportovaná ukázková prezentace se nachází na adrese <http://publikace.php5.cz/>.